

### Ejercicio 1.

Eres parte del equipo que está diseñando la caché para el procesador de un submarino no tripulado. Actualmente, el sistema tiene como primer nivel de caché una caché set-associative con líneas de 4 palabras, en la que datos y tags se acceden en paralelo. Si acceder a los tags demanda 2.5ns, acceder a los datos 4ns, determinar si hubo hit o miss demanda 1ns, y enviar el dato seleccionado al procesador en caso de hit demanda 1ns:

- ¿Cuáles son las latencias en caso de hit y de miss en caché?
- ¿Cuál sería la latencia de hit en caché si tanto acceder al tag como al dato demandaran 3ns?
- ¿Cuál es la cota superior a la frecuencia del procesador?

Se le ha agregado un siguiente nivel de caché, implementado con una memoria DRAM cuya latencia es, para la primera palabra, de 100ns y 10ns para cada palabra adicional accedida secuencialmente. Se está evaluando qué política de escritura usar. Si el 25% de las líneas de caché son modificadas (en promedio) 5 veces antes de ser sacadas de caché:

- ¿Cuál es el tiempo promedio de llevar una línea a caché si está implementada como write-back? ¿Y si fuera write-through?
- ¿Qué política de escritura demanda menos tiempo en escribir los datos en el próximo nivel de la caché?
- ¿Cuántas veces debería escribirse (en promedio) una línea para que la respuesta a la pregunta (b) cambie?

1) **Latencia Total** = Latencia acceder a Tag + Latencia acceder a datos + determinar hit o miss + enviar dato al procesador (en caso de hit)

Como el enunciado dice que los datos y tags **se acceden en forma paralela**, sabemos que solo se tardará 4ns en acceder a datos y tags.

Por ende:

a) LatenciaHit = Latencia acceder a Tag y Datos + determinar hit o miss + enviar dato al procesador (en caso de hit)

$$\text{LatenciaHit} = 4\text{ns} + 1\text{ns} + 1\text{ns} = \mathbf{6\text{ns}}$$

b) LatenciaMiss = Latencia acceder a Tag y Datos + determinar hit o miss =

$$\text{LatenciaMiss} = 4\text{ns} + 1\text{ns} = \mathbf{5\text{ns}}$$

b) Ahora acceder al tag y dato cuesta 3ns.

Pero siguen ocurriendo en forma paralela, por lo tanto no cambia mucho.

LatenciaHit = Latencia acceder a Tag y Datos + determinar hit o miss + enviar dato al procesador (en caso de hit)

$$\text{LatenciaHit} = 3\text{ns} + 1\text{ns} + 1\text{ns} = \mathbf{5\text{ns}}$$

c) Cota superior (el caso de hit ya que es en el paso en el que más tardamos, es decir, el paso que más tiempo utiliza el procesador)

$$\text{Cota Superior} = 1/6\text{ns} = \mathbf{0.16666\text{ ghz}}$$

## Ejercicio 2.

Se están analizando dos posibles microarquitecturas M1 y M2 para el mismo set de instrucciones. La frecuencia de M1 es 4GHz y el período del reloj de M2 es 0.4ns. Cada una tiene los siguientes CPI para cada tipo de instrucción:

Operations	M1	M2
Load-store	2	4
ALU	1	1
Branches	3	1

La operación del sistema se basa en la ejecución de tres rutinas: A, B y C. La rutina C se implementa combinando las 2 anteriores. La proporción de cada instrucción en las rutinas A y B es la siguiente:

Operations	A	B
Load-store	10%	20.00%
ALU	40%	50.00%
Branches	50%	30%

Si la implementación del programa C requiere ejecutar la misma cantidad de veces A y B, y del total de la ejecución las tres rutinas, el 60% corresponden a A, el 10% corresponden a B y lo restante a C, ¿qué microarquitectura recomendaría al equipo? Justifique.

### Para la arquitectura M1 tenemos:

- Período de reloj =  $1 / \text{frecuencia} = 1 / 4 \text{ GHz} = 0,25 \text{ ns}$
- A  $\rightarrow (2 * 0.10 + 1 * 0.40 + 3 * 0.5) = 2.1$
- B  $\rightarrow (2 * 0.2 + 1 * 0.5 + 3 * 0.3) = 1.8$
- C  $\rightarrow (0.5 * A + 0.5 * B) = 1.95$

### Para la arquitectura M2 tenemos:

- Período de reloj =  $0,4 \text{ ns}$
- A  $\rightarrow (4 * 0.10 + 1 * 0.40 + 1 * 0.5) = 1.3$
- B  $\rightarrow (4 * 0.2 + 1 * 0.5 + 1 * 0.3) = 1.6$
- C  $\rightarrow (0.5 * A + 0.5 * B) = 1.45$

Se sabe que para la rutina C tenemos:

- 60% de instrucciones de A
- 10% de instrucciones de B
- 30% de instrucciones de C

Veamos qué nos queda para cada microarquitectura:

Para la arquitectura M1:

- 60% de instrucciones de A =  $0.6 * 2.1 = 1.26$
- 10% de instrucciones de B =  $0.1 * 1.8 = 0.18$
- 30% de instrucciones de C =  $0.3 * 1.95 = 0.585$

**TOTAL** =  $(1.26 + 0.18 + 0.585) * \text{período de reloj} = 2.025 * 0.25 = \mathbf{0.50625}$

Para la arquitectura M2:

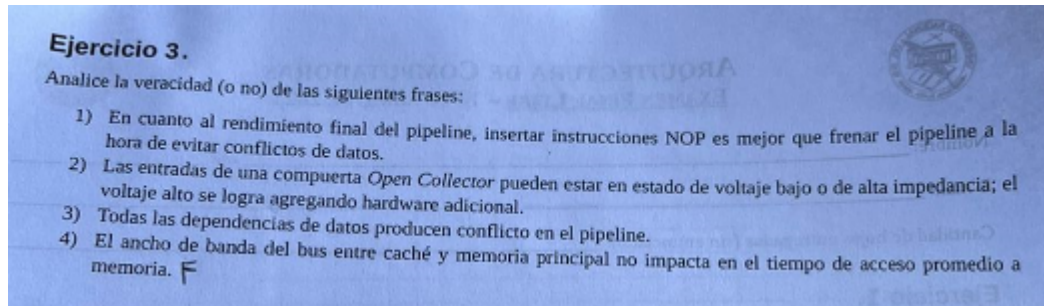
60% de instrucciones de A =  $0.6 * 1.3 = 0.78$

10% de instrucciones de B =  $0.1 * 1.6 = 0.16$

30% de instrucciones de C =  $0.3 * 1.45 = 0.435$

**TOTAL** =  $(0.78 + 0.16 + 0.435) * \text{período de reloj} = 1.375 * 0.4 = \mathbf{0.55}$

Por lo tanto, recomendaría el uso de la arquitectura M1 ya que utiliza menos Ciclos de reloj.



Ejercicio 4:

Pedía implementar:

- a) Un full adder y un sumador binario de 2 bits, independientes
- b) Un XOR de 4 bits

Utilizando solamente un sumador binario de 4 bits.

b)

